Data consistency and Handling Errors in Food Purchase Transactions

Analysis for handling below conditions:

* Cancelling order in case of overbooking – Etags can help to some extent. Compensating transactions can also take care of this scenario.
* Ensuring order transaction completes successfully.
* And cancelling the transaction if any of the operations involved in a transaction fails.

E.g.

Payment is successful but marking item as sold in the inventory errored. In this case all operations should be reversed or compensated (AKA Compensating Transactions).

1. Payment should be rolled back,
2. Update to Inventory, kitchen, Fridge/POS should roll back,
3. Order Status should be marked as failed
4. user should be notified of cancelled order

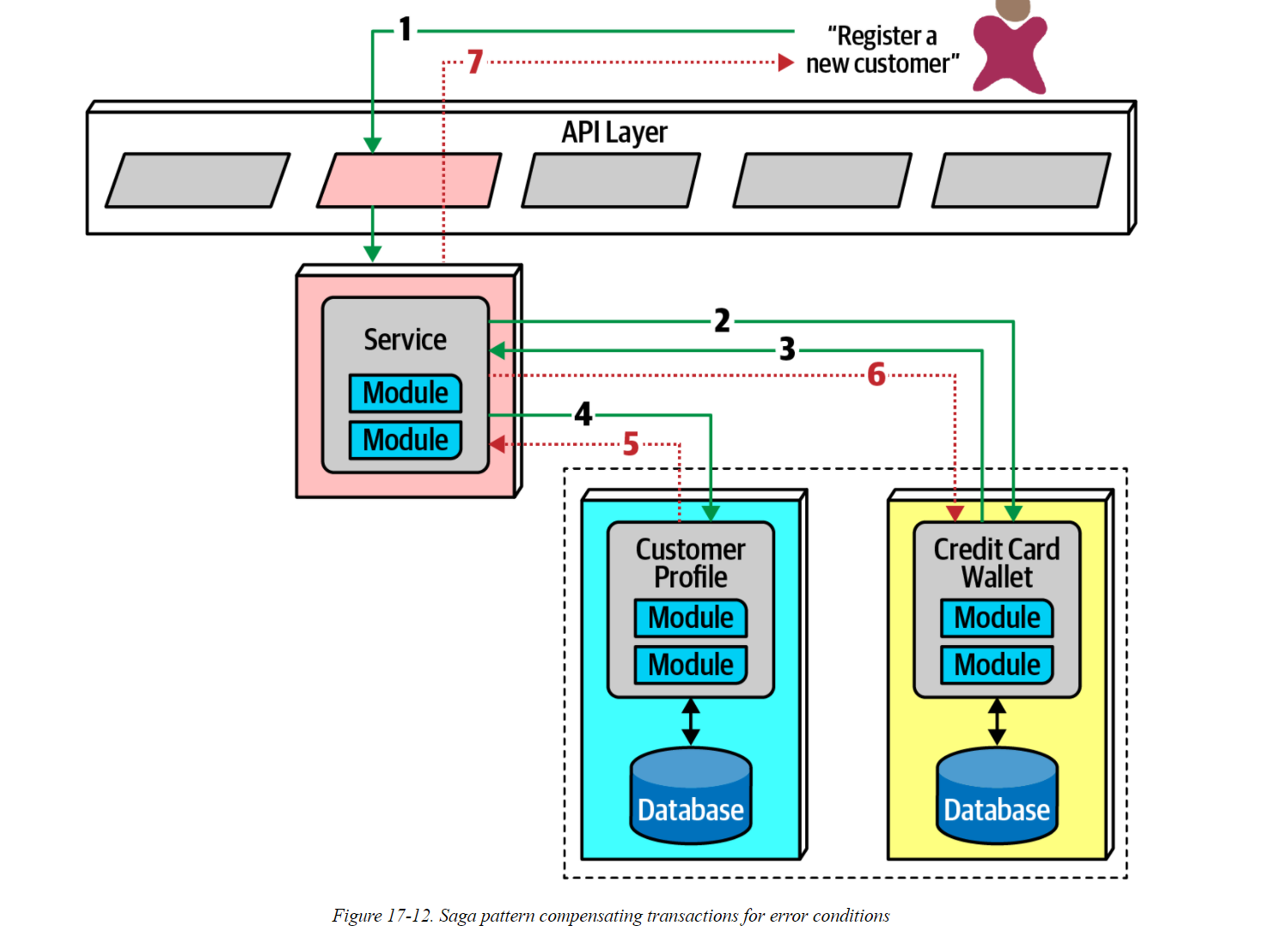
Note: I am assuming we will reserve the food at checkout and before the payment has been made and then mark the food as sold once the payment is successful.

**Ways to compensate: Saga Pattern in Microservices**

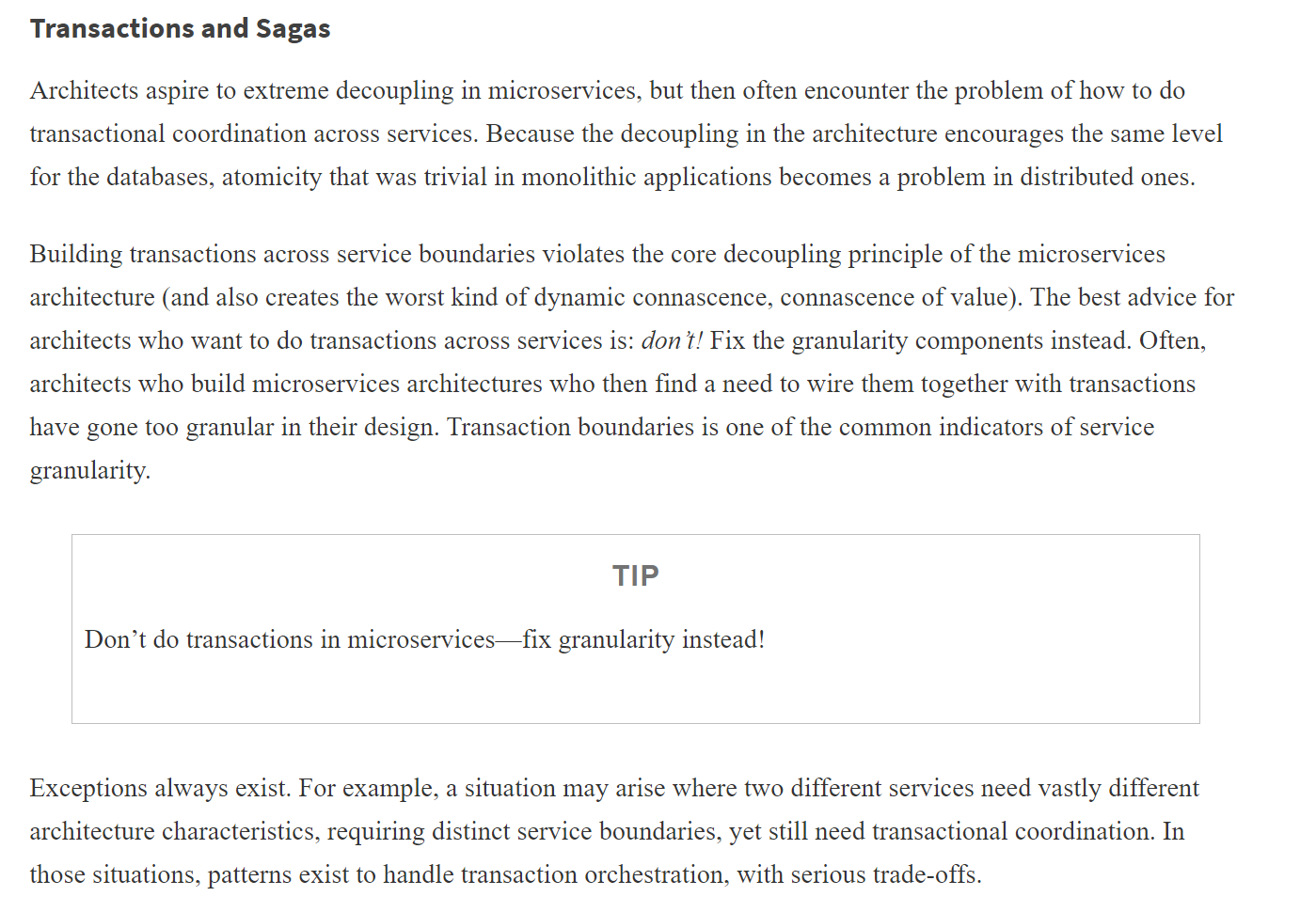
Via API Calls – For services with Synchronous communication. E.g. OrderService in our diagram can Call UndoPayment API etc. -Strong consistency – slow response to user

Via broadcast – For Services with asynchronous communication. Like Kitchen update, UndoReserveFood in fridge, user notification etc.

e.g. Payment and inventory has no direct communication so in case of payment failure, Payment can broadcast a failure transaction. -– Eventual consistency



**Point to Ponder – From – Fundamentals of Software Architecture book**



**Conclusion:**

We can decide combination of sync and asynch communication between these Services.

We can evaluate Microservice vs Service based architecture or some other arch style for order workflow.